

SYSTEM TEST DESCRIPTION TEMPLATE

Version 1.0

November 8, 2000

This page intentionally left blank

FOREWORD

This document was developed to provide any project developing regression or using production software with a template for generating a System Test Description (STD). This template should be supplemented with project-specific information to produce a STD that accurately describes the project's plan for testing the software.

This template describes the necessary elements to be considered when preparing a System Test Description. It is in accordance with the Military Standard (MIL-STD)-498 Data Item Description (DID) for an STD, Data Item (DI)- Information Processing Standards for Computers (IPSC)-81439.

Department of Education, Student Financial Assistance (SFA), will maintain and keep this STD template. Users of this document may report deficiencies and or corrections using the format provided in the Document Change Request (DCR) form. Department of Education, SFA, collects and processes reported information as inputs for process improvements to the STD template.

NOTE: *The use of automated techniques is encouraged. The term "document" used in this report means a collection of data regardless of its medium. Diagrams, tables, matrices, and other presentation styles are acceptable substitutes for text when data required by this report can be made more readable using these styles.*

This page intentionally left blank

GLOSSARY

Aggregate

A mass of distinct things gathered into a total or whole.

Aggregation Level

Effective measurement analysis and reporting requires that the data be aggregated to higher levels of the of the software components and project organizational structure. The aggregation levels define the different ways the measurement data can be grouped and organized for reporting on the project. The aggregation levels describe how the measurement data relates to an existing product and process structures. The organization that allows the measurement results to be combined, and later decomposed, into meaningful pieces of information.

Aggregation Structure

The structure used to define the data according to the defined aggregation levels. The levels may describe the personnel and management structure of the project, or the configuration of physical components of the project. All entries in a structure should be of the same type, such as software modules. However, these entries may reside at various levels of the structure, such as software modules at the unit level, CSCI, or integrated level of the software architecture.

Application

(1.) A complete, self-contained program that performs specific function(s) directly for the user.

(2.) In the TPM process this term refers to one of the two basic measurement activities which comprise the system measurement process. The application activity involves collecting, analyzing, and acting upon the measurement data.

See **Tailoring**

Automated Test Script

A computer readable set of instructions that performs a sequence of steps, sub-steps, or other actions, performed serially, in parallel, or in some combination of consecution, that creates the desired test conditions that the test case is deigned to evaluate.

Baseline

A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

Baseline Control

Baseline control is the process that regulates approved and released versions of all software, documentation, and the test environment throughout the test life cycle.

Black Box Testing

This is testing associated with functional testing where the object being tested is treated as a black box. In this type of testing the test object is subjected to inputs and outputs that are verified for conformance to prescribed specifications.

Capacity Testing

Attempts to simulate expected customer peak load operations in order to ensure that the system performance requirements are met. It does not necessarily exercise all of the functional areas of the system, but selects a subset that is easy to replicate in volume. It will ensure that functions which are expected to use the most system resources are adequately represented.

Change Control

The process by which problems and changes to the software, documentation, and test environment are evaluated, approved, rejected, scheduled, and tracked.

Computer Aided Software Engineering (CASE)

A technique for using computers to help with one or more phases of the software life cycle, including the systematic analysis, design, implementation and maintenance of software. Adopting the CASE approach to building and maintaining systems involves software tools and training for the developers who will use them.

Computer Software Configuration Item (CSCI)

An aggregation of software that is designated for configuration management and treated as a single entity in the configuration management process.

Configuration Control

An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification.

Configuration Item (CI)

Hardware or software, or an aggregate of both, which is designated by the project configuration manager (or contracting agency) for configuration management.

Configuration Management (CM)

A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements.

Configuration Management Office (CMO)

The Configuration Management Office (CMO) is the officiator of the project life cycle CM process.

Criteria

A standard, rules, or tests by which something can be judged.

Critical Defect

See Criticality

Criticality

The assessment of the impact upon a system of a given error, defect, problem, or discrepancy during the life cycle of a system.

The definition of critical and non-critical system defects or problems should be addressed at a management level and can be different for each system. For any given system error, defect, problem, or discrepancy, an appropriate impact value (i.e., priority) will be assigned.

An example of impact values with the corresponding priority numbers is presented below as contained in IEEE/EIA Std-12207, 1998. The priority that will apply if a problem can result in one or more of these impacts:

PRIORITY	IMPACT
1.	a.) Prevent the accomplishment of an operational or mission essential capability.
	b.) Jeopardize safety.
	c.) Cause significant technical, cost, or schedule risks to the project or to life cycle support of the system.
2.	a.) Adversely affect the accomplishment of an operational or mission essential capability and no work-around solution is known.
	b.) Adversely affect technical, cost, or schedule risks to the project or to life cycle support of the system, and no work-around is known.

3.
 - a.) Adversely affect the accomplishment of an operational or mission essential capability, but a work-around solution is known.
 - b.) Adversely affect technical, cost, or schedule risks to the project or to life cycle support of the system, but a work-around is known.
4.
 - a.) Results in user/operator inconvenience or annoyance, but does not affect a required operational or mission essential capability.
 - b.) Results in inconvenience or annoyance for development or support personnel, but does not prevent the accomplishment of the responsibilities of these personnel.
5.
 - a.) This priority denotes any other effect.

Customer

The organization that procures software systems for itself or another organization.

Developer

An organization that develops software products. The term “develop” may include develop, modification, integration, reengineering, sustaining engineering, maintenance, or any other activity that results in software products. The developer may be a contractor or a government agency.

Discrepancy

An inconsistency or disagreement found during testing between the actual and expected test results.

Document

A data medium and the data recorded on it that generally has permanence and can be read by a human operator or machine. Often used to describe human readable items only (e.g., technical documents, design documents, requirements documents, etc.).

Documentation

- (1.) A collection of documents on a given subject.
- (2.) The management of documents, that includes the actions of identifying, acquiring, processing, storing, and disseminating.
- (3.) Any written or pictorial information describing, defining, specifying, reporting or certifying activities, requirements, procedures, or results.

Driver

A software program that exercises a system or system component by simulating the activity of a higher level component.

Emulation

One system is said to emulate another when it performs in exactly the same way, though perhaps not at the same speed. A typical example would be the emulation of one computer by (a program running on) another. You might use emulation, as a replacement for a system whereas you would use a simulation if you just wanted to analyze it and make predications about it.

Emulator

Hardware or software that performs emulation.

Entry Criteria

A set of decision making guidelines used to determine whether a system under test is ready to move into, or enter, a particular phase of testing. Entry criteria tend to become more rigorous as the test phases progress.

Environment

The infrastructure in which a system is executing, consisting of hardware, operating system software, interfaces, etc.

Exit criteria

A set of decision-making guidelines used to determine whether a system under test is ready to exit a particular phase of testing. When exit criteria are met, either the system under test moves on to the next test phase or the test project is considered complete. Exit criteria tend to become more rigorous as the test phases progress.

Final System Test Report (FSTR)

Used to determine whether system testing is completed and to assure that software is ready for production.

Hardware Configuration Item (HWCI)

An aggregation of hardware that is designated for configuration management and treated as a single entity in the configuration management process.

Independent Verification and Validation (IVV)

The verification and validation of a software product by an organization that is both technically and managerially separate from the organization responsible for developing the product.

Indicator

A measure or combination of measures that provides insight into a system issue or concept. TPM frequently uses indicators that are comparisons, such as planned versus actual measures. Indicators are generally presented as graphs or tables.

Integration

Combining software or hardware components or both into an overall system.

Integration Testing

The period of time in the software lifecycle during which the application is tested in a simulated production environment to validate the communications and technical architecture of the system. This test phase occurs when all the constituent components of the system under test are being integrated.

Interactive Development Environment (IDE)

A system for supporting the process of writing software. Such a system may include a syntax-directed editor, graphical tools for program entry, and integrated support for compiling and running the program and relating compilation errors back to the source code.

Interface

- (1.) A shared boundary (e.g., a hardware component linking two devices or registers, or a portion of storage accessed and/or modified by two or more computer programs).
- (2.) To interact or communicate with another system component.

Interface Requirement

A requirement that specifies a hardware, software, or database element with which a system or system component must interface, or that sets forth constraints caused by such an interface.

Interface Specification

A specification that sets forth the interface requirements for a system or system component (e.g., the software interface specification document).

Interface Testing

Tests conducted to ensure that program or system components correctly pass data and/or control to one another.

Issue

An area of concern where obstacles to achieving program objectives might arise. Issues include risks, problems, and lack of information. These three types of issues are defined as:

- Risk -- An area of concern that could occur, but is not certain. A risk is a potential problem. Risks represent the potential for the realization of unwanted, negative consequences from a project event. For example, a project plan may be based on the assumption that a COTS component will be available on a given date. There is a possibility (probability) that the COTS may be delayed and have some amount of negative impact on the project.
- Problem -- An area of concern that a project is currently experiencing or is relatively certain to experience. For example, a shortage of staff with the right skills may be an actual problem that is delaying the project.
- Lack of Information -- An area where the available information is inadequate to reliably predict project impact. Thus, satisfaction of project objectives is questionable even if no problems or risks are present. For example, lack of information about the size of the software to be developed could result in the project “discovering” that it has more work to do than originally planned.

Measure

The result of counting or otherwise quantifying characteristics of a process or product. Measures are numerical values assigned to system attributes according to defined criteria.

Measured (or actual) Value

Actual, current measurement data, such as hours of effort expended or line of code produced.

Measurement

The process of assigning quantitative values of system properties, according to some defined criteria. This process can be based on estimation or direct measurement. Estimation defines planned or expected measures. Direct measurement results in actual measures.

Measurement Analysis

The uses of measurement data to identify problems, assess problem impact, project an outcome, or evaluate alternatives related to system issues.

Measurement Analyst

The person(s) or team responsible for tailoring and applying system measures for a given project or task.

Measurement Information

Knowledge derived from analysis of measurement data and measurement indicators.

Milestone

A scheduled event for which some project or task member or manager is held accountable. A milestone is often used to measure progress.

Module

A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading.

Note: *The terms 'module', 'component', and 'unit' are often used interchangeably or defined to be sub-elements of one another in different ways depending on the context.*

Non-Critical Defect

See Criticality

Performance Testing

The period of time in the system or software development lifecycle during which the response times for the application are validated to be acceptable. The tests ensure that the system environment will support production volumes, both batch and on-line.

Priority

A measure of the elements of importance related to the repair of a system problem that are not considered in defining the severity of a system problem.

Project Manager (PM)

The official responsible for acquiring, developing, or supporting a system to meet technical, cost, schedule, and quality requirements. Acquisition, development, and support will include both internal tasks and work that is contracted to another source.

Quality Assurance (QA)

A planned and systematic pattern of all actions necessary to provide adequate confidence that the product optimally fulfils customers expectations.

Quality Control (QC)

The assessment of product compliance. Independently finding deficiencies assures compliance of the product with stated requirements.

Requirement

- (1.) A condition or capability needed to solve a problem or achieve an objective.
- (2.) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. The set of all requirements forms the basis of development.

Regression testing

Part of the test phase of software development where, as new modules are integrated into the system and the added functionality is tested, previously tested functionality is re-tested to assure that no new module has corrupted the system.

Risk

An area of concern that may occur, but is not certain. A risk is a potential problem. Risks represent the potential for the realization of unwanted, negative consequences from a project event. For example, a project plan may be based on the assumption that a commercial off the shelf (COTS) component will be available on a given date. There is a possibility (probability) that the COTS may be delayed and have some amount of negative impact on the project.

Severity

The degree to which a problem adversely influences the system's operation or the overall test effort.

Simulation

Attempting to predict aspects of the behavior of a system by creating an approximate (mathematical) model of it. This can be done by physical modeling, by writing a special-purpose computer program or using a more general simulation package, aimed at a particular kind of simulation. Typical examples are aircraft simulators or electronic circuit simulators.

Simulator

Hardware or software that performs simulation.

Software Design Specification (SDS)

A document that records the design of a system or system component; typical contents include: system and/or component algorithms, control logic, data structures, data set use, input/output formats, and interface descriptions.

Software Development File (SDF)

The developer shall document the development of each Computer Software Unit (CSU), Computer Software Component (CSC), and CSCI in Software Development Files (SDF). The developer shall establish a separate SDF for each CSU or a logically related group of CSUs, for each CSC or a logically related group of CSCs, and for each CSCI. The developer shall document and implement procedures to establish and maintain SDFs. SDFs may be generated, maintained, and controlled by automated means. To reduce duplication, SDFs should not contain information provided in other documents or SDFs. The set of SDFs shall include (directly or by reference) the following information:

- Design considerations and constraints.
- Design documentation and data.
- Scheduling and status information.
- Test requirements and responsibilities.
- Test case, test case procedures, and results.

Software Life Cycle

The phases a software product goes through between when it is conceived and when it is no longer available for use. The software life cycle typically includes the following: requirements, analysis, design, construction, testing (validation), installation, operation, maintenance, and retirement. The development process tends to run iteratively through these phases rather than linearly; several models (spirals, waterfall, etc.) have been proposed to describe this process.

Other processes associated with a software product are: quality assurance, marketing, sales, and support.

Software Management Plan

A project plan for the development of the software component of a system or for the development of a software product.

Software Requirements Document (SRD)

This is a formal document derived from the Software Requirements Specification (SRS) that sets forth the requirements, specifications, and standards for a system (e.g., a software product). Typically included are functional specifications and requirements, performance specifications and requirements, interface specifications and requirements, design specifications and requirements, and development requirements and standards.

Software Requirements Specification (SRS)

A specification that sets forth the requirements for a system component; (e.g., a software product). Typically included are functional requirements, performance requirements, interface requirements, design requirements, and development standards.

Software Tool

Computer programs used to help develop, test, analyze, or maintain another computer program or its documentation.

Specification

Documentation containing a precise, detailed, verifiable description of particulars with respect to the requirements, design, function, behavior, construction, or other characteristics of a system or system component.

Stub

(1.) A dummy procedure used when linking a program with a run-time library. The stub routine need not contain any code and is only present to prevent “undefined label” errors at link time.

(2.) A local procedure in a remote procedure call (RPC). The client calls the stub to perform some task and need not necessarily be aware that RPC is involved. The stub transmits parameters over the network to the server and returns the results to the client/caller.

System

(1.) Any large program.

(2.) The entire computer system, including the input/output devices, supervisor program or operating system and possibly other software.

System Problem Report (SPR)

A form that is used to record a discrepancy discovered during the Integration Test, Performance Test and System Qualification Test phases of the SI&T process concerning a Computer Software Configuration Item, a software system or subsystem, other software related items, and associated documentation.

System Problem Report (SPR) Status Report

The System Problem Report Status Report is used during the SPR Status Review to determine if the SPRs are being processed appropriately and expeditiously.

System Testing

The period of time in the software lifecycle during which the implementation of each requirement is validated.

Tailoring

In the TPM process, this term refers to one of the two basic measurement activities, which comprise the system measurement process. The tailoring activity includes identification and prioritization of program issues, selection and specification of appropriate system measures, and integration of the measurement requirements to the developer's system process.

See **Application**.

Test

The process of exercising a product to identify differences between expected and actual behavior.

Test Artifacts

An item created during the system integration and test process that is preserved upon completion of the test process (e.g., test plans, requirements documentation, automated test scripts, and test documentation).

Test Case

A description of a test to be executed for or focused on a specific test aim.

Test Case Procedures

A sequence of steps, sub-steps, and other actions, performed serially, in parallel, or in some combination of consecution, that creates the desired test conditions that the test case is designed to evaluate.

Test Case (Setup) Suite

The steps required to configure the test environment for execution of a test case.

Testing Condition

System state or circumstance created by proceeding through some combination of steps, sub-steps, or actions in a test case.

Testing Environment

The infrastructure in which the test is performed, consisting of hardware, system software, test tools, and procedures.

Test Plan

In a test plan the general structure and the strategic choices with respect to the test to be executed are formulated. The test plan forms the scope of reference during execution of the test and also serves as an instrument to communicate with the customer of the test. The test plan is a description of the test project, including a description of the activities and planning, therefore it is *not* a description of the tests themselves.

Test Readiness Review (TRR)

Review conducted to determine whether a software test phase has been completed and to assure that the software is prepared for the next step in the formal integration and testing procedures. Software test procedures and results are evaluated, for compliance with the software testing requirements and system descriptions, for adequacy in accomplishing testing goals. Also, provides the forum for updating and revising operational and supporting documentation.

Test Resources

Aids that are used by a test tool for collecting, tracking and controlling information. This information is:

- Software requirements defined in the Software Requirements Document.
- Test requirements defined in the System Test Description.
- Automated test case scripts as defined in the System Test Description.

- SPRs as determined at each phase of the System Integration and Testing process.

This information is controlled by Configuration Management at the end of the SI&T process for use whenever further testing may be conducted, using a testing tool, during the remaining lifecycle of the software or system.

Test Tools

The software, hardware, systems, or other instruments that are used to measure and test an item.

Traceability

Degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor, successor, or master-subordinate relationship to one another (e.g., the degree to which the requirements and design of a given software component match).

Unit

The lowest element of a software hierarchy that contains one or more of the following characteristics:

- A unit comprising one or more logical functional entities.
- An element specified in the design of a computer software component that is separately testable.
- The lowest level to which software requirements can be traced.
- The design and coding of any unit can be accomplished by a single individual within the assigned schedule.

Unit Test

The process of ensuring that the unit executes as intended. This usually involves testing all statements and branch possibilities.

Version

One of a sequence of copies of a system, each incorporates new modifications.

Version Identifier

A unique identifier assigned to baseline software, documentation, and test environment.

Version Control

The process by which all changes to the software, documentation, and test environment are compiled and built into a new version of the system.

Version Control Report

A report that details all changes and enhancements made to current version of the software, documentation, and test environment.

White Box Testing

This type of testing is associated with structural testing in which the testing can be characterized as being tied to implementation details, such as control methods, database design, coding details, and logic paths. The process of how an individual input is treated to produce a given output is ascertained. Structural testing is sometimes referred to as “clear box testing” since white boxes are considered opaque and do not really permit visibility into the code.

Work Breakdown Structure (WBS)

A work breakdown structure for software defines the software-related elements associated with program work, work activities, and products. Many measures are aggregated and analyzed at various WBS levels.

This page intentionally left blank

DOCUMENT CHANGE REQUEST

DOCUMENT: System Test Description (STD) Template, Version 1.0

SFA TRACKING NUMBER: _____

NAME OF SUBMITTING ORGANIZATION: _____

ORGANIZATION CONTACT: _____ TELEPHONE: _____

MAILING ADDRESS: _____

DATE: _____ SHORT TITLE: _____

CHANGE LOCATION: _____

(Use section #, figure #, table #, etc.)

PROPOSED CHANGE:

REASON FOR CHANGE:

.....
Note: For the Department of Education, Office of Student Financial Assistance (SFA), to take appropriate action on a change request, please provide a clear description of the recommended change along with supporting reason.

Send to: Department of Education, Office of Student Financial Assistance, , 400 Maryland Ave. SW, Washington DC 20202 or Fax to: (202) 205-8532

This page intentionally left blank

RECORD OF CHANGES

CHANGE NUMBER	DATE	FIGURE, TABLE, OR PARAGRAPH NUMBER	A/M/D*	TITLE OR BRIEF DESCRIPTION	CHANGE REQUEST NUMBER

* **A** = ADD
 M = MODIFY
 D = DELETE

This page intentionally left blank

DOCUMENT CONVENTIONS

[Text]

Replace text.

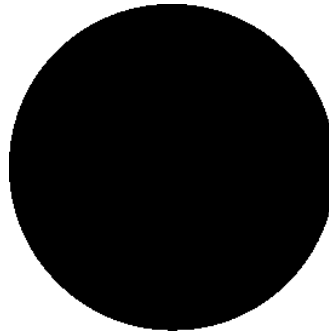
text in italics

Notes or instructions to the author.

NOTE: *The next page is the start of the template that begins with a project or system STD title and approval page. Delete this page and preceding pages in the final format of your project or system STD. Remember to update the header page to reflect the submission date and your document configuration identifier for the project or system STD.*

This page intentionally left blank

**SYSTEM TEST DESCRIPTION
FOR
[PROJECT NAME]**



[Submission Date]

**Prepared for:
Department of Education
Student Financial Assistance
400 Maryland Ave. SW
Washington DC 20202**

**Prepared by:
[Organization Name and Address]**

[Project Name] System Test Description
Version [x.y]
[Submission Date]

This page intentionally left blank

SYSTEM TEST DESCRIPTION FOR [PROJECT NAME]

The undersigned certify that they have reviewed the System Test Description (STD). It has been determined that this deliverable meets requirements, meets requirements with concerns, or does not meet requirements. If the deliverable meets requirements with concerns, or does not meet requirements, an explanation must be provided along with necessary documentation.

Software Project Manager Signature

Date _____

- ☐ Meets Requirements
 - ☐ Meets Requirements with Concerns
 - ☐ Does Not Meet Requirements
- _____
- _____

Integration Testing Team Manager Signature

Date _____

- ☐ Meets Requirements
 - ☐ Meets Requirements with Concerns
 - ☐ Does Not Meet Requirements
- _____
- _____

[Project Name] System Test Description

Version [x.y]

[Submission Date]

Integration Testing Team Leader Signature

Date _____

- ☐ Meets Requirements
- ☐ Meets Requirements with Concerns
- ☐ Does Not Meet Requirements

[Additional signatures may be appended]

RECORD OF CHANGES

VERSION NUMBER	DATE	DESCRIPTION
1.0	November 2000	Original release of document

This page intentionally left blank

TABLE OF CONTENTS

1. SCOPE	1-1
1.1 IDENTIFICATION	1-1
1.2 SYSTEM OVERVIEW	1-1
1.3 DOCUMENT OVERVIEW	1-1
2. REFERENCED DOCUMENTS	2-1
2.1 GOVERNMENT DOCUMENTS	2-1
2.2 [NON-GOVERNMENT DOCUMENTS]	2-1
3. TEST PREPARATIONS	3-1
3.1 [PROJECT-UNIQUE IDENTIFIER OF A TEST]	3-1
3.1.1 Hardware Preparation	3-1
3.1.2 Software Preparation.....	3-1
3.1.3 Other Pre-Test Preparations.....	3-1
4. TEST DESCRIPTIONS	4-1
4.1 [PROJECT-UNIQUE IDENTIFIER OF A TEST]	4-2
4.1.1 [Project-Unique Identifier Of A Test Case].....	4-2
4.1.1.1 Requirements Addressed	4-2
4.1.1.2 Prerequisite Conditions.....	4-3
4.1.1.3 Test Inputs.....	4-3
4.1.1.4 Expected Test Results.....	4-3
4.1.1.5 Criteria For Evaluating Results.....	4-4
4.1.1.6 Test Case Procedure.....	4-5
4.1.1.7 Assumptions and Constraints.....	4-6
5. REQUIREMENTS TRACEABILITY	5-1
6. NOTES	6-1
6.1 ABBREVIATIONS AND ACRONYMS	6-1
APPENDIX A. [ADDITIONAL DATA]	A-1

This page intentionally left blank

1. SCOPE

1.1 Identification

This paragraph shall contain a full identification of the system and the software to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).

1.2 System Overview

This paragraph shall briefly state the purpose of the system and the software to which this document applies. It shall describe the general nature of the system and software; summarize the history of system development, operation, and maintenance; identify the project sponsor, acquirer, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.

1.3 Document Overview

This paragraph shall summarize the purpose and contents of this document and shall describe any security or privacy considerations associated with its use.

This page intentionally left blank

2. REFERENCED DOCUMENTS

This section shall list the number, title, revision, and date of all documents referenced in this report. This section shall also identify the source for all documents not available through normal Government stocking activities.

2.1 Government Documents

- a. MIL-STD-498, DID, DI-IPSC-81438
- b. Systems Engineering Process Office (SEPO), Naval Command Control & Oceanographic Surveillance Center (NCCOSC), Software Test Plan Template Ver. 1.2
- c. [Add government documents referenced by your project]

2.2 [Non-Government Documents]

- (a) [Add non-government documents referenced by your project]

This page intentionally left blank

3. TEST PREPARATIONS

This section shall be divided into the following paragraphs. Safety precautions shall be marked by WARNING or CAUTION and security and privacy considerations shall be included as applicable.

3.1 [Project-Unique Identifier Of A Test]

This paragraph shall identify the test by project-unique identifier, shall provide a brief description, and shall be divided into the following subparagraphs. When the information required duplicates information previously specified for another test, that information may be referenced rather than repeated.

3.1.1 Hardware Preparation

This paragraph shall identify and describe procedures necessary to prepare the hardware for the test. Reference may be made to published operating manuals for these procedures. The following shall be provided, as applicable:

- a. The specific hardware to be used. Identify the hardware by name and, if applicable, model number.*
- b. Any switch settings and cabling necessary to connect the hardware.*
- c. One or more diagrams to show hardware, interconnecting control, and data paths.*
- d. Step-by-step instructions for placing the hardware in a state of readiness.*

3.1.2 Software Preparation

This paragraph shall describe the procedures necessary to prepare the item(s) under test and any related software, including data, for the test. Reference may be made to published operating manuals for these procedures. The following shall be provided, as applicable:

- a. The specific software to be used in the test.*
- b. The storage medium of the item(s) under test (e.g., magnetic tape, diskette, hard disk, etc.).*
- c. The storage medium of any related software (e.g., simulators, test drivers, databases, etc.).*
- d. Instructions for loading the software, including required sequence.*
- e. Instructions for software initialization common to more than one test case.*

3.1.3 Other Pre-Test Preparations

This paragraph shall describe other pre-test personnel actions, preparations, or procedures necessary to perform the test.

This page intentionally left blank

4. TEST DESCRIPTIONS

*This section shall be divided into the following paragraphs to identify and describe each test to which this STD applies. Safety precautions shall be marked by **WARNING** or **CAUTION** and security and privacy consideration shall be included as applicable.*

Multiple planned tests can be described in this section and, therefore, an expansion of this section will occur. The following is an example of how the additional section numbering is to be implemented using “variable indexing”.

<i>Template Section Number</i>	<i>Description</i>	<i>Variable Indexing</i>
<i>4.1</i>	<i>[Project-Unique Identifier of a Test]</i>	<i>4.W</i>
<i>4.1.1</i>	<i>[Project-Unique Identifier of a Test Case]</i>	<i>4.W.X</i>
<i>4.1.1.1</i>	<i>Requirements Addressed</i>	<i>4.W.X.Y</i>
<i>4.1.1.2</i>	<i>Prerequisite Conditions</i>	<i>4.W.X.Y+1</i>
<i>4.1.1.3</i>	<i>Test Inputs</i>	<i>4.W.X.Y+2</i>
<i>4.1.1.4</i>	<i>Expected Test Results</i>	<i>4.W.X.Y+3</i>
<i>4.1.1.5</i>	<i>Criteria For Evaluating Results</i>	<i>4.W.X.Y+4</i>
<i>4.1.1.6</i>	<i>Test Case Procedures</i>	<i>4.W.X.Y+5</i>
<i>4.1.1.6.1</i>	<i>Individual Test Case Procedure</i>	<i>4.W.X.Y+5.Z</i>
<i>4.1.1.7</i>	<i>Assumptions and Constraints</i>	<i>4.W.X.Y+6</i>

For this example of using variable indexing for the STD, the following assumptions are made: Two tests, one test has two test cases and the other test has one test case. The two test cases each have two test case procedures and the other test case has three test case procedures.

<i>Template Section Number</i>	<i>Description</i>	<i>Variable Indexing</i>
<i>4.1</i>	<i>Project-Unique Identifier of Test #1</i>	<i>4.W</i>
<i>4.1.1</i>	<i>Project-Unique Identifier of Test Case #1</i>	<i>4.W.X</i>
<i>4.1.1.1</i>	<i>Requirements Addressed</i>	<i>4.W.X.Y</i>
<i>4.1.1.2</i>	<i>Prerequisite Conditions</i>	<i>4.W.X.Y+1</i>
<i>4.1.1.3</i>	<i>Test Inputs</i>	<i>4.W.X.Y+2</i>
<i>4.1.1.4</i>	<i>Expected Test Results</i>	<i>4.W.X.Y+3</i>
<i>4.1.1.5</i>	<i>Criteria For Evaluating Results</i>	<i>4.W.X.Y+4</i>
<i>4.1.1.6</i>	<i>Test Case Procedures</i>	<i>4.W.X.Y+5</i>
<i>4.1.1.6.1</i>	<i>Test Case Procedure #1</i>	<i>4.W.X.Y+5.Z</i>
<i>4.1.1.6.2</i>	<i>Test Case Procedure #2</i>	<i>4.W.X.Y+5.Z+1</i>
<i>4.1.1.7</i>	<i>Assumptions and Constraints</i>	<i>4.W.X.Y+6</i>
<i>4.1.2</i>	<i>Project-Unique Identifier of Test Case #2</i>	<i>4.W.X+1</i>
<i>4.1.2.1</i>	<i>Requirements Addressed</i>	<i>4.W.X+1.Y</i>
<i>4.1.2.2</i>	<i>Prerequisite Conditions</i>	<i>4.W.X+1.Y+1</i>
<i>4.1.2.3</i>	<i>Test Inputs</i>	<i>4.W.X+1.Y+2</i>

4.1.2.4	<i>Expected Test Results</i>	$4.W.X+1.Y+3$
4.1.2.5	<i>Criteria For Evaluating Results</i>	$4.W.X+1.Y+4$
4.1.2.6	<i>Test Case Procedures</i>	$4.W.X+1.Y+5$
4.1.2.6.1	<i>Test Case Procedure #1</i>	$4.W.X+1.Y+5.Z$
4.1.2.6.2	<i>Test Case Procedure #2</i>	$4.W.X+1.Y+5.Z+1$
4.1.2.7	<i>Assumptions and Constraints</i>	$4.W.X+1.Y+6$
4.2	<i>Project-Unique Identifier of Test #2</i>	$4.W+1$
4.2.1	<i>Project-Unique Identifier of Test Case #1</i>	$4.W.X$
4.2.1.1	<i>Requirements Addressed</i>	$4.W+1.X.Y$
4.2.1.2	<i>Prerequisite Conditions</i>	$4.W+1.X.Y+1$
4.2.1.3	<i>Test Inputs</i>	$4.W+1.X.Y+2$
4.2.1.4	<i>Expected Test Results</i>	$4.W+1.X.Y+3$
4.2.1.5	<i>Criteria For Evaluating Results</i>	$4.W+1.X.Y+4$
4.2.1.6	<i>Test Case Procedures</i>	$4.W+1.X.Y+5$
4.2.1.6.1	<i>Test Case Procedure #1</i>	$4.W+1.X.Y+5.Z$
4.2.1.6.2	<i>Test Case Procedure #2</i>	$4.W+1.X.Y+5.Z+1$
4.2.1.6.3	<i>Test Case Procedure #3</i>	$4.W+1.X.Y+5.Z+2$
4.2.1.7	<i>Assumptions and Constraints</i>	$4.W+1.X.Y+6$

4.1 [Project-Unique Identifier Of A Test]

This paragraph shall identify a test by project-unique identifier and shall be divided into the following subparagraphs. When the information required duplicates information previously provided, that information maybe referenced rather than repeated.

4.1.1 [Project-Unique Identifier Of A Test Case]

This paragraph shall identify a test case by project-unique identifier, state its purpose, and provide a brief description. The following subparagraphs shall provide a detailed description of the test case.

4.1.1.1 Requirements Addressed

This paragraph shall identify the CSCI or system requirements addressed by the test case. (Alternatively, this information may be provided in Section 5.a of this document)

4.1.1.2 Prerequisite Conditions

This paragraph shall identify any prerequisite conditions that must be established prior to performing the test case. The following considerations shall be discussed, as applicable:

- a. Hardware and software configuration.
- b. Flag, initial breakpoints, pointers, control parameters of initial data to be set/reset prior to commencing test.
- c. Preset hardware conditions or electrical states necessary to run the test case.
- d. Initial conditions to be used in making timing measurements.
- e. Conditioning of the simulated environment.
- f. Other special conditions peculiar to the test case.

4.1.1.3 Test Inputs

This paragraph shall describe the test inputs necessary for the test case. The following shall be provided, as applicable:

- a. Name, purpose, and description (e.g., range of values, accuracy, etc.) for each test input.
- b. Source of the test input and the method to be used for selecting the test input.
- c. Whether the test input is real or simulated.
- d. Time or event sequence of test input.
- e. The manner in which the input data will be controlled to:
 1. Test the item(s) with a minimum/reasonable number of data types and values.
 2. Exercise the item(s) with a range of valid data types and values that test for overload, saturation, and other “worst case” effects.
 3. Exercise the item(s) with invalid data types and values to test for appropriate handling of irregular inputs.
 4. Permit re-testing, if necessary.

4.1.1.4 Expected Test Results

This paragraph shall identify all expected test results for the test case. Both intermediate and final test results shall be provided, as applicable.

4.1.1.5 Criteria For Evaluating Results

This paragraph shall identify the criteria to be used for evaluating the intermediate and final results of the test case. For each test result the following information shall be provided, as applicable:

- a. The range or accuracy over which an output can vary and still be acceptable.*
- b. Minimum number of combinations or alternatives of input and output conditions that constitute an acceptable test result.*
- c. Minimum/maximum allowable test duration, in terms of time or number of events.*
- d. Maximum number of interrupts, halts, or other system breaks that may occur.*
- e. Allowable severity of processing errors.*
- f. Conditions under which the result is inconclusive and re-testing is to be performed.*
- g. Conditions under which the outputs are to be interpreted as indicating irregularities in input test data, in the test database, or data files or in test procedures.*
- h. Allowable indications of the control, status, and results of the test and the readiness for the next test case (e.g., may be the output of auxiliary test software).*
- i. Additional criteria not mentioned above.*

4.1.1.6 Test Case Procedure

This paragraph shall define the test case procedure(s) for the test case as a series of individually numbered steps listed sequentially in the order in which the steps are to be performed for the test case. For convenience in document maintenance, the test case procedures may be included as an appendix and referenced in this paragraph. The appropriate level of detail in each test case procedure depends on the type of software being tested. For some software, each keystroke may be a separate test case procedure step; for most software, each step will include a logically related series of keystrokes or other actions. The appropriate level of detail is the level which is useful to specify expected results and compare them to the actual results. The following shall be provided for each test case procedure, as applicable:

a. Test operator actions and equipment operation required for each step, including commands, as applicable, to:

- 1. Initiate the test case and apply test results.*
- 2. Inspect test conditions.*
- 3. Perform interim evaluations of test results.*
- 4. Record data.*
- 5. Halt and interrupt the test case.*
- 6. Request data dumps or other aids, if needed.*
- 7. Modify the database or data files.*
- 8. Repeat the test case if unsuccessful.*
- 9. Apply alternate modes as required by the test case.*
- 10. Terminate the test case.*

b. Expected result and evaluation criteria for each step.

c. If the test case addresses multiple requirements, identification of which test procedure step(s) address which requirements. (Alternatively, this information may be provided in Section 5 of this document.)

d. Actions to follow in the event of a program stop or indicated error, such as:

- 1. Recording of critical data from indicators for reference purposes.*
- 2. Halting or pausing time-sensitive test support software and test apparatus.*
- 3. Collection of system and operator records of test results.*

e. Procedures to be used to reduce and analyze test results to accomplish the following, as applicable:

- 1. Detect whether an output has been produced.*
- 2. Identify media and location of data produced by the test case.*
- 3. Evaluate output as a basis for continuation of test sequence.*
- 4. Evaluate test output against required output.*

4.1.1.7 Assumptions and Constraints

This paragraph shall identify any assumptions made and constraints or limitations imposed in the descriptions of the test case due to system or test conditions, such as limitations on timing, interfaces, equipment, personnel, and database or data files. If waivers or exceptions to specified limits and parameters are approved, they shall be identified and this paragraph shall address their effect(s) and impact(s) upon the test case.

5. REQUIREMENTS TRACEABILITY

This paragraph shall contain:

- a. Traceability from each test case in this STD to the system or CSCI requirements it addresses. If a test case addresses multiple requirements, traceability from each set of test procedure steps to the requirement(s) addressed. (Alternatively, this traceability may be provided in 4.2.1.1.)*
- b. Traceability from each system or CSCI requirement covered by this STD to the test case(s) that address it. For CSCI testing, traceability from each CSCI requirement in the CSCI Software Requirements Specification (SRS) and associated Interface Requirements Specification (IRS). For system testing, traceability from each system requirement in the system System/Subsystem Specification (SSS) and associated IRSs. If a test case addresses multiple requirements, the traceability shall indicate the particular test procedure steps that address each requirement.*

This page intentionally left blank

6. NOTES

This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, and rationale). This section shall include an alphabetic listing of all acronyms and abbreviations and their meaning as used in this document and a list of any terms and definitions needed to understand this document.

6.1 Abbreviations And Acronyms

LIST OF ACRONYMS	
CI	Configuration Item
CM	Configuration Management
CMO	Configuration Management Office
DCR	Document Change Request
DI	Data Item
DID	Data Item Description
CSCI	Computer Software Configuration Item
HWCI	Hardware Configuration Item
IRS	Interface Requirements Specification
MIL-STD	Military Standard
NCCOSC	Naval Command Control & Oceanographic Surveillance Center
QA	Quality Assurance
QC	Quality Control
RPC	Remote Procedure Call
SDP	Software Development Plan
SEPO	Systems Engineering Process Office
SFA	Office of Student Financial Assistance
SI&T	System Integration and Testing
SPR	System Problem Report
SRD	Software Requirements Document
SRS	Software Requirements Specification
SSS	System/Subsystem Specifications
STD	System Test Description
STP	System Test Plan
STR	System Test Report
TRR	Test Readiness Review

This page intentionally left blank

APPENDIX A. [ADDITIONAL DATA]

Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally be provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

This page intentionally left blank